

# UC3M at Da-Vincis-2023: using BETO for Detection of Aggressive and Violent Incidents on Social Networks

Jorge Luis Saavedra Rubio<sup>1,†</sup>, Alejandro Valbuena Almeida<sup>1,†</sup> and Isabel Segura-Bedmar<sup>1,\*</sup>

<sup>1</sup>Universidad Carlos III de Madrid, Av. Universidad 30, Leganes, Madrid - Spain (28911)

## Abstract

This article presents the contribution of the UC3M\_PLN team to the DA-VINCIS competition organized at IberLEF-2023. The objective of the task is to identify accidents, murders, robberies, and other incidents on Twitter. The DA-VINCIS Corpus has been created using various messages recovered from tweets associated with violent acts. The competition includes subtasks for the identification and multi-categorization of violent events. We have successfully completed both subtasks by exploring different strategies to combine transformer language features and embeddings.

## Keywords

Natural Language Processing, Spanish-BERT, BETO, RoBERTa, ALBETO, DistilBERT, Aggressive and Violent Classification

## 1. Introduction

Violence has clear negative impacts on those who witness or experience it, such as increased rates of depression, anxiety, post-traumatic stress disorder, and more. Additionally, acts of violence greatly affect governments as they are responsible for ensuring the safety of their population [8]. Therefore, the identification and monitoring of violent incidents are of utmost importance.

In this context, social media platforms serve as valuable sources of information for detecting and tracking violent events. People frequently post real-time updates about such incidents, providing an excellent opportunity for researchers in the field of information technology. By leveraging Natural Language Processing (NLP) techniques, researchers can develop solutions that enable timely identification of violent occurrences in social networks [2]. These solutions can assist authorities in responding more efficiently to real-time events and formulating crime prevention strategies based on geographical locations and event types. Moreover, these solutions would also prove beneficial to the general population, as they can stay informed about ongoing violent events in real-time and their specific locations [4].

---

*IberLEF 2023, September 2023, Jaén, Spain*

\*Corresponding author.

†These authors contributed equally.


✉ jlsaavedra89r@gmail.com (J. L. S. Rubio); alejandrov90@gmail.com (A. V. Almeida); isegura@inf.uc3m.es (I. Segura-Bedmar)

🆔 0000-0001-5937-1840 (J. L. S. Rubio); 0009-0008-8887-4699 (A. V. Almeida); 0000-0002-7810-2360

(I. Segura-Bedmar)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

The shared task "DA-VINCIS: Detection of Aggressive and Violent Incidents from Social Networks in Spanish" [10] organized at IberLEF 2023, aims to promote the development of automatic models for determining whether news obtained from Twitter describes a violent act using both texts and images. In 2022, there was a previous edition of this shared task [3] that also comprised two tracks: a binary classification task to determine whether tweets were associated with violent incidents or not, and a multi-label classification task to identify the category of the violent incident. Transformer-based solutions were found to achieve competitive results in both subtasks [11, 13]. These works demonstrate the effectiveness of leveraging pre-trained transformers for violence detection and classification in social media data. In 2023, the shared task proposes two subtasks: the identification of violent events and the recognition of categories of violent events. using images and text. In this way, unlike the first edition, the objective of the task is to promote the development of multimodal methods to detect tweets describing a violent event.

This paper describes the participation of UC3M\_PLN in both subtasks of the DA-VINCIS shared task. Although the current edition of the shared task aims to promote the development of multimodal methods to detect tweets describing a violent event, due to our lack of experience in the task, we decided to only focus on texts. Thus, our team's strategy for the identification and classification of aggressive and violent events is based on fitting and combining Transformers with linguistic features. In recent years there have been considerable advances in contextualized language models [6, 12], where non-English language versions have also been made available [9, 1, 14]. Due to their increasing use, many lightweight versions of these models with reduced parameters have also been released to speed up training and inference times. However, versions of these lighter models (e.g., ALBERT , DistilBERT [2]) for languages other than English are still scarce.

One of the challenging aspects of the task lies in handling the inherent complexity and ambiguity present in incident classification [3]. Incidents can exhibit significant variations in their nature and context, which can pose potential difficulties in accurately categorizing them. To tackle this challenge, we leverage the strong language understanding capabilities and contextual embeddings of the BERT model [6]. This enables us to capture nuanced patterns and contextual cues that assist in precise classification. Furthermore, by fine-tuning the pre-trained BERT model specifically for the classification task at hand, our system can adapt to the domain intricacies and enhance its performance. Another significant challenge involves effectively dealing with the presence of noisy or irrelevant features in the data. These features have the potential to introduce unnecessary complexity and hinder the model's generalization ability. To address this issue, our system incorporates preprocessing steps, such as removing punctuation marks, hyperlinks, and emojis, as well as correcting spelling errors and expanding acronyms and abbreviations. These preprocessing techniques play a crucial role in reducing noise within the data, thereby providing cleaner inputs to the model. Consequently, this improves the model's capability to capture meaningful patterns and make accurate predictions.

## 2. Task Overview

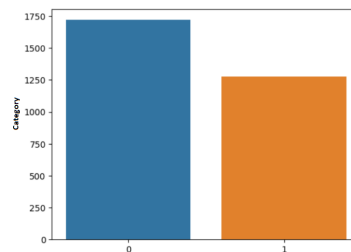
The DA-VINCIS task aims to identify the presence of violent incidents on Twitter and classify them into specific crime categories. The task consists of two subtasks:

**Subtask 1:** Violent event identification, which involves determining whether a given tweet is related to a violent incident or not.

**Subtask 2:** Violent event category recognition, which focuses on classifying the type of crime into predefined categories, including:

- **Accident:** Refers to an eventual event or action that results in involuntary damage to people or things.
- **Murder :** Involves the intentional deprivation of life.
- **Robbery:** Involves the seizure or destruction of other people’s property without right and without the consent of those who can legally dispose of them.
- **Other:** This category is selected when the tweet is not associated with any of the previous categories. It includes tweets about other crimes such as kidnapping and tweets that do not report any crime.

To facilitate the task, a dataset comprising 2,996 Spanish tweets related to violent acts is provided. The dataset is slightly unbalanced for the binary classification problem (violent vs. non-violent) (see Fig. 1) and unbalanced for the multiclass environment (accident, murder, robbery, other) (see Fig. 2). The organizers have provided separate training and test sets. Moreover, we selected a split from the training set for validation purposes. The split was created using stratified sampling to maintain label balance. The distribution of the classes is similar in the data subsets (see Fig. 3).

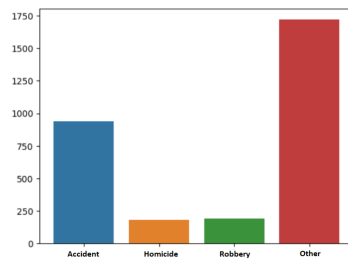


**Figure 1:** Data distribution with labels for Task Binary Classification

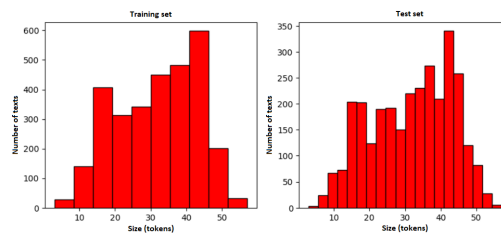
In terms of evaluation measures, the F1 measure of the violent class is used to assess the performance of participating systems in subtask 1, while in subtask 2, the Macro-F1 score is chosen as the evaluation metric.

## 3. System Configuration

In this section, we present the system configuration and methodology employed in the binary and multilabel classification tasks. The following subsections outline the key steps involved in the system implementation and the algorithms utilized.



**Figure 2:** Class distribution for the subtask 2



**Figure 3:** Distribution of the size of the texts

### 3.1. Binary Classification

To begin with, the dataset was divided into a training set consisting of 2,247 samples and a validation set containing 749 samples. A refined version of the dataset was obtained by performing several preprocessing steps. These steps involved removing punctuation marks, hyperlinks, and emojis, as well as correcting spelling errors and expanding acronyms and abbreviations. A tweet is associated with a violent or non-violent incident in a binary classification task. As shown in Figure 1, this task has training data with two labels violent incident(0) and nonviolent incident(1).

The HuggingFace's transformers library was utilized for text tokenization in Spanish, employing the "dccuchile/bert-base-spanish-wwm-cased" model. The tokenizer was used to convert each text into a numerical representation based on tokens. Special tokens such as [CLS] (sequence start) and [SEP] (sequence separator) were added, and the maximum length was limited to 128 tokens. Texts were truncated or padded as necessary, and attention masks were generated. The tokenization results, i.e., token IDs and attention masks, were stored in separate lists, namely `input_ids` and `attention_mask`. Preprocessing functions were employed to tokenize the training (`X_train`) and validation (`X_val`) datasets.

A PyTorch DataLoader was created from the input data (`x_inputs`, `x_masks`) and corresponding labels (`y_labels`). The DataLoader was utilized to load data in batches during model training and validation. The `batch_size` parameter was set to 32, indicating that each batch would contain 32 data samples. This facilitated dividing the data into smaller batches and processing them in parallel, thereby enhancing training efficiency. PyTorch tensors, `y_train_labels`, and `y_val_labels`, were created from the training and validation labels (`y_train` and `y_val`, respec-

tively). These tensors were used as input for the DataLoader. A TensorDataset object was created, combining the input data (x\_inputs and x\_masks) and labels (y\_labels) into a single dataset. Additionally, a sequential sampler (SequentialSampler) was defined to ensure that data was retrieved sequentially during training. Finally, a DataLoader was instantiated using the dataset, sampler, and batch size (batch\_size), with num\_workers set to 0 to utilize a single data loading thread. The result was two DataLoaders: train\_dataloader containing training data and val\_dataloader containing validation data, both ready for model training.

To ensure reproducibility, a seed was set for different random number generators in Python and PyTorch. This ensured consistent results whenever the code was executed using the same seed. Reproducibility and result comparison were facilitated for operations involving random numbers, such as weight initialization in machine learning models, random data splitting, or number generation in experiments.

For the classification task, a sequence classification model based on the pre-trained Spanish BERT model called "dccuchile/bert-base-spanish-wwm-cased" was chosen. This model has been specifically trained on Spanish text and exhibits strong language understanding capabilities. The BERT model [6] is well-suited for a wide range of natural language processing tasks, including text classification [7]. BERT is an encoder trained using two strategies: masked language modeling (MLM) and next sentence prediction (NSP). In our case, the model is configured for binary classification, with num\_labels=2, as we are aiming to classify incidents into two categories.

The model's parameters, including output\_attentions and output\_hidden\_states, are set to False, indicating that the model does not return attention weights and hidden states in its output. To optimize the model's parameters, the AdamW optimizer is utilized. It leverages the Adam algorithm with weight decay regularization to update the model's parameters. The optimizer employs the model's parameters (model.parameters()) and sets the learning rate (lr) to 4e-5 and the epsilon term (eps) to 1e-6, empirically.

The training process involves multiple epochs, each consisting of iterations over the training data. The total number of training steps (total\_steps) is calculated by multiplying the length of the training dataloader (train\_data loader) by the number of epochs. To adjust the learning rate during training, a learning rate scheduler (scheduler) is created using the get\_linear\_schedule\_with\_warmup method. This scheduler starts with a warm-up period (num\_warmup\_steps = 0) and applies linear scheduling to adjust the learning rate over the training steps (num\_training\_steps = total\_steps) [12].

After each training epoch, the model is validated to assess its performance on unseen data. The model is switched to evaluation mode (model.eval()), and the validation dataloader (validation\_dataloader) is iterated over. The predictions of the model for each batch are compared with the actual labels to compute the temporary evaluation accuracy (tmp\_eval\_accuracy). The overall evaluation accuracy is obtained by accumulating the temporary accuracies and dividing by the total number of batches[14].

### **3.2. Multilabel Classification**

First, tweets were preprocessed as we described in the previous subsection.

We utilize the BETO tokenizer to convert the text messages into input token sequences and attention masks.

The BERT pre-trained model is instantiated for sequence classification into four different categories. The arguments `output_attentions=False` and `output_hidden_states=False` are set to configure the model to exclude attentions and hidden states during inference. An AdamW optimizer is defined with a specific learning rate. The number of training epochs is set, and a learning rate scheduler is created using the `get_linear_schedule_with_warmup` method. This scheduler automatically adjusts the learning rate during training, with `num_warmup_steps=0` indicating no warm-up phase. The `num_training_steps` is set to the previously calculated total training steps. These components are utilized during the model training process.

During the training, the loss function, `BCEWithLogitsLoss`, is chosen for multilabel classification problems. The accuracy metric is defined, specifying `average='macro'` to calculate the average precision for all classes, `task="multiclass"` to indicate a multiclass classification problem, `num_classes=4` to specify the number of classes, and `mdmc_average='samplewise'` to calculate classification accuracy for multiple samples. The forward pass is performed through the model using `outputs = model(b_input_ids, token_type_ids=None, attention_mask=b_input_mask)`.

The model's output passes through a sigmoid function and is rounded to obtain the final prediction. The function returns the prediction as an array of integers, representing the predicted labels for each message in the test set. These predictions are stored for further analysis.

## 4. Results

Our results using BETO are shown in Tables 1 and 2 for the subtasks 1 and 2, respectively. These tables also show the results of the other participating teams. The results were published by the organizers and are sorted in descending order based on the F1-score values.

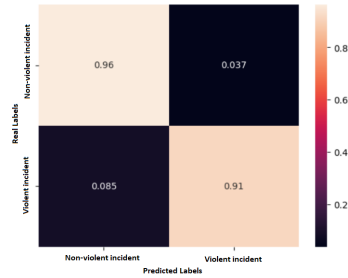
**Table 1**

Comparison of the results for subtask 1. Our results are in bold.

Team	Precision	Recall	F1-score
danielvallejo237	0.9302	0.9226	0.9264
EstebanPonce	0.9006	0.9409	0.9203
<b>UC3M</b>	<b>0.9067</b>	<b>0.9308</b>	<b>0.9186</b>
agmegias	0.8951	0.9389	0.9165
csuazob	0.8939	0.9267	0.9100
Arnold	0.9014	0.9124	0.9069
rkcd	0.9000	0.8982	0.8991
VickBat	0.9081	0.8859	0.8969
HoracioJarquin	0.9456	0.8493	0.8948

Regarding the subtask 2, our system demonstrates competitive performance in recognizing categories of violent acts (see Table 4). As it happened in subtask 1, our system again ranks the third position, with scores very close to the best system.

Currently, we do not know what approaches have been used by the other teams. Although



**Figure 4:** Confusion matrix for the binary classification

**Table 2**

Comparison of the results for subtask 2 with macro scores. Our results are in bold.

Team	Precision	Recall	F1-score
EstebanPonce	0.8737	0.8864	0.8797
agmegias	0.8523	0.8973	0.8733
<b>UC3M</b>	<b>0.8622</b>	<b>0.8784</b>	<b>0.8698</b>
Arnold	0.8305	0.8715	0.8492
csuazob	0.8441	0.8577	0.8490
HoracioJarquin	0.7663	0.9407	0.8427
danielvallejo237	0.8394	0.8449	0.8421
BrauuHdzm	0.8027	0.8091	0.8048
escom	0.8178	0.7974	0.8036

our approach is unimodal (that is, we have not exploited the images in tweets), our results are similar to the other teams.

These results effectively highlight the potential of pre-trained linguistic models that are specifically designed for the Spanish language. They showcase promising and competitive outcomes in the identification and categorization of violent events.

## 5. Conclusions

In this article, we have detailed our team’s participation in the DA-VINCIS task, which focuses on the identification and categorization of violent events on social media platforms. Both subtasks were approached by exploring two strategies: initialization and configuration of Transformer model embeddings. Our team ranked third in subtask 1 (F1 score of 0.9186) and third in subtask 2 (F1 score of 0.8698). It is important to note that our results are biased due to our custom validation. Therefore, better strategies and configurations for selecting pretrained models should be evaluated. Additionally, the performance in the second subtask is limited and needs to be improved, as there are some categories where any instance has been correctly classified. Furthermore, the dataset provided for the competition was slightly unbalanced, with imbalances in both the binary classification problem (violent vs. non-violent) and the multiclass environment (accident, murder, robbery, other). Balancing techniques, such as oversampling or

undersampling, could be applied to address this issue and improve the model's performance on minority classes. In terms of data preprocessing, our system employed various techniques to reduce noise and improve the quality of input data. However, there is room for further exploration and refinement of these preprocessing steps. For example, exploring the impact of different text normalization techniques or incorporating more advanced techniques, such as sentiment analysis, could potentially enhance the model's ability to accurately classify violent events. Future work should focus on incorporating external knowledge sources, such as domain-specific ontologies or lexicons, to improve the model's understanding and classification of violent incidents. These additional resources could provide valuable context and semantic information that can aid in better categorization and identification of violent events on social networks. By addressing the aforementioned points, we believe that our system's performance can be further enhanced, ultimately contributing to the timely identification and monitoring of violent incidents on social media platforms.

## 6. Online Resources

The sources for related work are available via Jupyter Notebook <https://github.com/jlsaavedra/DA-VINCIS-2023>

## 7. Citations and Bibliographies

### References

- [1] García-Díaz, J. A., Jiménez-Zafra, S. M., Rodríguez-García, M. Á., Valencia-García, R. (2022). UMUTeam at DA-VINCIS 2022: Aggressive and Violent Classification using Knowledge Integration and Ensemble Learning.
- [2] S. Stieglitz, M. Mirbabaie, B. Ross, C. Neuberger, Social media analytics – Challenges in topic discovery, data collection, and data preparation, *International Journal of Information Management* 39 (2018) 156–168. doi:<https://doi.org/10.1016/j.ijinfomgt.2017.12.002>.
- [3] L. J. Arellano, H. Jair Escalante, L. Villaseñor-Pineda, M. Montes y Gómez, F. Sánchez-Vega, Overview of DA-VINCIS at IberLEF 2022: Detection of Aggressive and Violent Incidents from Social Media in Spanish, *Procesamiento del Lenguaje Natural* 69 (2022). J. A. García-Díaz, R. Valencia-García, Compilation and evaluation of the Spanish
- [4] S. M. Jiménez-Zafra, R. Morante, E. Blanco, M. T. Martín-Valdivia, L. A. Ureña-López, Detecting negation cues and scopes in Spanish, in: *Proceedings of The 12th Language Resources and Evaluation Conference*, 2020.
- [5] É. Grave, P. Bojanowski, P. Gupta, A. Joulin, T. Mikolov, Learning Word Vectors for 157 Languages, in: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.
- [6] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 1.



- [7] T. Davidson, D. Warmley, M. Macy, I. Weber, Automated hate speech detection and the problem of offensive language, in: Proceedings of the International AAAI Conference on Web and Social Media.
- [8] S. Madisetty, M. S. Desarkar, Aggression detection in social media using deep neural networks, in: Proceedings of the first workshop on trolling, aggression and cyberbullying (TRAC-2018), 2018.
- [9] J. Cañete, S. Donoso, F. Bravo-Marquez, A. Carvallo, V. Araujo, Albeto and distilbeto: Lightweight spanish language models, arXiv preprint arXiv:2204.09145 (2022).
- [10] H. Jarquín-Vásquez, D. Irazú Hernández Farías, J. Arellano, H. Jair Escalante, L. Villaseñor-Pineda, M. Montes y Gómez, F. Sanchez-Vega. Overview of DA-VINCIS at IberLEF 2023: Detection of Aggressive and Violent Incidents from Social Media in Spanish. *Procesamiento del Lenguaje Natural*, vol 71, septiembre 2023.
- [11] García-Díaz, J. A., S. M. Jiménez-Zafra, M. Rodríguez-García, and R. Valencia-García. 2022. UMUTeam at DA-VINCIS 2022: Aggressive and Violent classification using Knowledge Integration and Ensemble Learning. In Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2022), CEUR Workshop Proceedings.
- [12] Montañes-Salas, R. M., R. del Hoyo-Alonso, and P. Peña-Larena. 2022. ITAINNOVA@DA-VINCIS: A Tale of Transformers and Simple Optimization Techniques. In Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2022), CEUR Workshop Proceedings.
- [13] Tonja, A. L., M. Arif, O. Kolesnikova, A. Gelbukh, and G. Sidorov. 2022. Detection of Aggressive and Violent Incidents from Social Media in Spanish using Pretrained Language Model. In Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2022), CEUR.
- [14] A. Gutiérrez-Fandiño, J. Armengol-Estapé, M. Pámies, J. Llop-Palao, J. Silveira-Ocampo, C. P. Carrino, C. Armentano-Oller, C. Rodríguez-Penagos, A. Gonzalez-Agirre, M. Villegas, Maria: Spanish language models, *Procesamiento del Lenguaje Natural*.